Tiny URL

Given a (typically) long URL, how would how would you design service that would generate a shorter and unique alias for it.

Discuss things like:

How to generate a unique ID for each URL?

How would you generate unique IDs at scale (thousands of URL shortening requests coming every second)?

How would your service handle redirect?

How would you support custom short URLs?

How to delete expired URLs etc?

How to track click stats?

API:

1. Create
   1. HTTPPOST
   2. Request: long URL
   3. Response: Short URL
2. Get
   1. HTTPGET
   2. Request: Short URL
   3. Response: Long URL
3. All the short URLs should be unique

Database:

1. ShortURL: Random string
2. LongURL: URL

Random string Algorithm:

1. A - Z: 26
2. a – z: 26
3. 0 – 9: 9
4. 62^7 🡪 3,521,614,606,208‬, 3 trillion

Scale:

1. For every 1 short url we have 100 redirections
2. On an average if we have 50 million users
3. 50 \* 100 – 5 billion redirections
4. 50 million / 30 days \* 24 hrs \* 60 min \* 60 sec 🡪 20 short urls per second
5. 5 billion / 30 days \* 24 hrs \* 60 min \* 60 sec 🡪 2000 short urls redirect per second
6. NoSQL database as we don’t need to maintain relationships to perform joins
7. NoSQL also helps in scalability
8. We also can cache the frequently used data

Key Generator:

1. Different service, hash it and convert it to base 64
2. We can have two instances, and each can store multiple keys into the database
3. We can delete the instance and give back the key
4. If one server goes down other can support

API:

1. Can request the key generator and verify it exist in DB, else store it
2. If key found return the response with 302 (Redirect) else 404 (Not Found)
3. Custom key: validate the length and special characters to keep it consistent with other keys or we can add a special character which makes it unique.

DB partitions

Cache implementation for fast retrieval

Load balancers to split the request between multiple servers

Cleaning the database

